1. Introduction: some history

Pencil was originally an application created by Patrick Corrieri by the end of 2005 under the name “Pencil Planner”. It was a simple yet effective pencil test (a.k.a. line test) programme. It included a drawing area, a single-track timeline where keyframes could be added, previous and next onion skins. The drawing tools were a pencil and an eraser, along with a colour chooser. The animation could be played at a specified frame rate, and in addition it was possible to add a background image and a background sound. It was developed under Qt 3 and made available on Macintosh and then Windows platforms under GPL license. The Mac version could output QuickTime files. There were two releases v0.1 and v0.2

When I became aware of this programme, I realised that it could be evolved into an application for making traditional animations rather than just pencil tests. Obviously what was needed was more drawing tools to fill shapes with colours, and some selection tools to move things around. I got in touch with Patrick, and started implementing those functions, along with cleaning the original code.

I also implemented tablet support. This became v0.3 of Pencil, released in October 2006. Pleased with the result, I went on with more ambitious goals. On the one hand, the basics of traditional animation is to sketch pencil tests, and then draw clean images on top of those sketches.

Another fundamental of traditional animation is to be able to organise separate elements (background, character, or parts of it) into different superimposed layers. This definitely called for the possibility of superimposing layers. On the other hand, I also convinced myself that vector graphics rather than raster (bitmap) graphics were useful for making animated drawings. Vector graphics can be rendered at any resolution, can easily be deformed and resized, colours can be readjusted, etc. They are quite well suited to render the traditional animation style: contours filled with some colours – inherited from the “cell(uloid)” style. Therefore I decided to introduce not only raster layers, but vector layers as well. Yet I wanted vector drawing to have the same artistic feel as raster drawing, so I implemented the same tools for both layers – although some of them such as the pencil and brush behaved differently.This lead to v0.4 released in January 2007.

It was a huge jump from the previous versions. A lot of the original software was rewritten at this stage. Although incomplete, buggy, and still very unsatisfactory at many levels as an animation programme, it had the capacity to produce decent animations.

At that time, Patrick told me that he was pleased to follow these developments but did not have the time or capacity to get further involved in them, so he kindly agreed to grant me administration of the project. This was perfect timing as I had already invested a lot of efforts (in particular on the vector side) and started developing a personal vision of what the application should be in its final version. I considered creating a new project with a new name, but failing to find anything better than “Pencil”, I kept the existing sourceforge project. I supplemented it with a newly designed website, a forum, and later and wiki to get more feedback and public attention.

From then, I brought many refinements (some of them being quite unknown such as curved gradients) and got some occasional help from people such as Benjamin Raverdy, and Mark Schmelzenbach, although I remained mostly the only developer. The latest version v0.4.4 to date (released in early 2008)

As of May 2010, this version is downloaded about 1000 times a day, according to the Sourceforge statistics.

After this version, I had to make a pause for several reasons. Firstly I could not dedicate as much time to Pencil, and secondly I had long realised that the code had grown organically and inherited many poorly designed elements and architectures which needed some cleaning. So I set myself to this task. Some time has passed. Fortunately, then came along Oluseyi Sonaiya who by and large shares the same vision of Pencil as me, and also decided to take advantage of his experience as a developer to clean and extend Pencil to the next level.

2. The spirit of this project

When I started to publicise Pencil, my intention was to make everyone benefit from my ongoing effort to design an animation application. However, while being an open-source project, I believe that an application like Pencil should be designed with some personal vision and goal to remain effective. I hope other developers will share my vision and join this effort, and that Pencil will become useful for most people interested in animation. As general rules,

* feedback and suggestions are very welcome and will get due consideration
* any kind of contribution is very welcome if it indeed integrates into the vision and concept of the project, and will get consideration for official release
* people interested in making substantial changes inconsistent with or departing from the concept of the project are free to do so, provided they create their own distinct project, or possibly subproject (optional extension, etc).

It does **not** mean

* that Pencil will necessarily comply to anyone’s needs, habits, desires.
* that the Pencil project is everyone’s, that anyone can do whatever they want with it, claim new version, or dismiss credits of people who contributed to it.

These points, in my opinion, follow common sense and the usual acceptance of the term open-source, consistent with the GPL license of Pencil. (If it is not the case, the license should be updated accordingly).

So far I have developed Pencil mainly by myself and for myself. Now that several people are interested in participating in the project in the long term, it has become necessary to provide some information about my original vision and concept of Pencil. This is the goal of this document. It was kindly suggested to me by Oluseyi.

3. The concept

3.1 Why this project?

As I once wrote on the Pencil website, I have always liked drawing and the art of creating motion from a series of drawings. As a child, I used to make animation by drawing pictures on each page of a notebook. It was fun and simple, but not so convenient and quite limited (compared to a real animation film featuring colours, sound, etc).

Displaying pictures and playing sound is an ability that computers quickly acquired. As a result, a lot of animation software naturally emerged over time. However the usual input systems (keyboard and mouse) used by computers, along with conventional software design, have kept those programmes remote from the original drawing experience of traditional animation. Fortunately, pen tablets and later tablet PC appeared and made it possible to get an experience much closer to actual drawing. Still I found that existing animation programmes were either

* not taking advantage of the tablet
* not recreating the drawing experience of traditional animation, or not completely satisfactory in their design
* oriented towards other goals than traditional animation
* not available / affordable to the general class of users

So I decided to develop an application which would be simple, available to anyone, and designed to recreate the drawing experience of traditional animation, along with some other benefits coming from using a computer. In a sense, it is an attempt at conciling the traditional pencil approach advocated by people such as [Hayao Miyazaki: drawn to perfection](http://www.telegraph.co.uk/culture/film/6911264/Hayao-Miyazaki-drawn-to-perfection.html) and the advantages offered by today’s computers, in particular pen tablets. This is the guiding line of the project.

3.2 What Pencil is not:

* Pencil is not and will never be a clone of some existing software
* Pencil is not and will never be defined as the “free of charge” alternative of some existing commercial software
* While Pencil might get inspiration from existing software, it is not required to follow closely the interface or design of some existing software just because it meets some users’ natural expectation. Original design/paradigm might prevail over previously established ones.
* For practical reasons, and unless some essential function makes it necessary, Pencil in its internal structure will not be bound to comply to some predefined format or protocol, other than those defined by the Qt framework. In particular, Pencil is not a SVG editor or Flash editor, etc. It should be however be able to import and export various useful formats with some reasonable accuracy/compliance to those formats’ specifications.

## **3.3 What Pencil is:**

The primary goal of Pencil is to create traditional animation, which consists in drawing several pictures to create the illusion of motion.

Because of its potential, it also appeared that it may be used in the future for related usages such as

* drawing, painting, editing pictures (without animating)
* drawing storyboards or comic strips
* taking notes/sketches/doodles and organising them
* editing videos.

However, these are not the primary goal of Pencil at present.

Finally, Pencil might be the basis for an integrated solution for animation project management, often referred to as “animation studio”.

In order to allow the secondary usages, Pencil will be designed in a very modular way from v0.5 and beyond.

## **3.4 The general interface**

### **3.4.1 Appearance**

Visually, Pencil should be dominated by the drawing/viewing area since it is focused on drawing.

Tools, options, and other technical elements should appear around this area in a non-obstrusive way and kept to a minimum. Their graphical representation should also assume a simple (but pleasing) appearance to let the user work with some “peace of mind”.

In that respect, the current Windows version has some unnecessary embossing lines and approximate layout which leaves an uncomfortable impression. The Mac version displays unnecessary boxes around each tool.

The platforms of reference will be pen tablet laptop computers and single-screen desktop/laptop computers, although we might also consider multi-screen usage. To satisfy different users, the interface will be customisable to a large degree, although predefined, thoroughly thought interface environments will be provided by default for each usage and platform.

Ideally, Pencil should be a multidocument application; open documents could be indicated by a series of tabs at the top of the screen.

to let the user work with some “peace of mind”.

In that respect, the current Windows version has some unnecessary embossing lines and approximate layout which leaves an uncomfortable impression. The Mac version displays unnecessary boxes around each tool.

The platforms of reference will be pen tablet laptop computers and single-screen desktop/laptop computers, although we might also consider multi-screen usage. To satisfy different users, the interface will be customisable to a large degree, although predefined, thoroughly thought interface environments will be provided by default for each usage and platform.

Ideally, Pencil should be a multidocument application; open documents could be indicated by a series of tabs at the top of the screen.

3.4.2 Multiple usages/environments

The current and only usage of Pencil is creating animations. When (or if) secondary usages become available, it is expected that Pencil will have some option to switch between those environments. It may appear as a menu in Pencil menu bar or as a series of buttons at the top of the screen (easier for tablet users), or even as a splash screen option.

The idea behind the different usage environments is to display (or possibly alter) only the elements which are relevant for a given usage.

For example, the current interface of Pencil is dedicated to drawing animations. It has the following elements : A layer/timeline palette displaying both layers and their associated tracks, a colour palette, a tool palette, a parameter/option palette, and a display palette. A menu bar (or a function bar with buttons when using a tablet) contains the basic file/edit functions (open, save, import, export, undo, redo) and some more specific functions.

This versatility is quite interesting, although it raises many questions.

Obviously the same kind of information cannot always be displayed in different environments (an animation is not quite a note which is not quite a picture). This would call for creating variants of the Pencil format for each usage, which would effectively correspond to having a suite of different applications just like OpenOffice. Another, possibly more fruitful approach is to allow the Pencil document format to have a section for each kind of environment. For example, the same file could contain a section of animation and a section of notes of handwritten ideas, instructions, or sketches which could be copied and pasted in the animation section. Of course, if a file contains only one section, it would open by default in the corresponding environment. In the case of creating a new document or opening a documents with several sections, a splash screen would ask the user which environment to load first.

Can this be done in practice? The danger of this approach is that Pencil become an overweight application. While it might be acceptable to wait several seconds to load a full-featured animation programme, it might be less so in the case of a quick image-editing programme or note-taking application. One possible solution would be a light-weight core that could dynamically load only those modules which are required for a given usage. The effectiveness of this approach should be investigated.

Other practical questions arise about the interface. For example, the onion-layer buttons allow to display/hide previous and next images in semi-transparency, which is a traditional useful trick to draw animation. It may be seen first as a display feature. However, it does not make sense in usages which do not involve time and the notion of previous and next image. In that sense, it should be included in the timeline (which disappears when switching to non-time related usages). On the other hand, it would hardly be useful in, say, video-editing mode. Defining where and when the elements appear obviously requires some thinking.

4. The animating environment

The following describes specifically the animation environment, which is Pencil’s current main goal and prority.

4.1 Drawing

4.1.1 Raster and Vector Layers

Pencil has two drawing modes: raster and vector. Although it is advisable for the user to know the difference between the two, it is essentially a technical difference about how drawings made by the user are encoded and stored in the programme. In most applications, the choice of one of these two technical implementations heavily influences the way the user can input and interact with the drawing. As a result, raster graphics programmes and vector graphics programmes usually have quite different interfaces and tools.

In Pencil, I want to break with this tradition and offer a common interface which is close and to the natural drawing experience. Other conventional tools more specific to raster or vector modes might be added as a special set for more advanced or conservative users.

There are many styles of 2D animations, from the conventional American celluloid style to the Russian sketch/paint style, and Pencil will give some freedom in that respect. Yet the celluloid style is the most common one and will be taken as a reference. We will assume the following steps:

* sketch/pencil/line tests
* draw clean lines and temporary lines
* fill areas with colours.